

FIG. 1

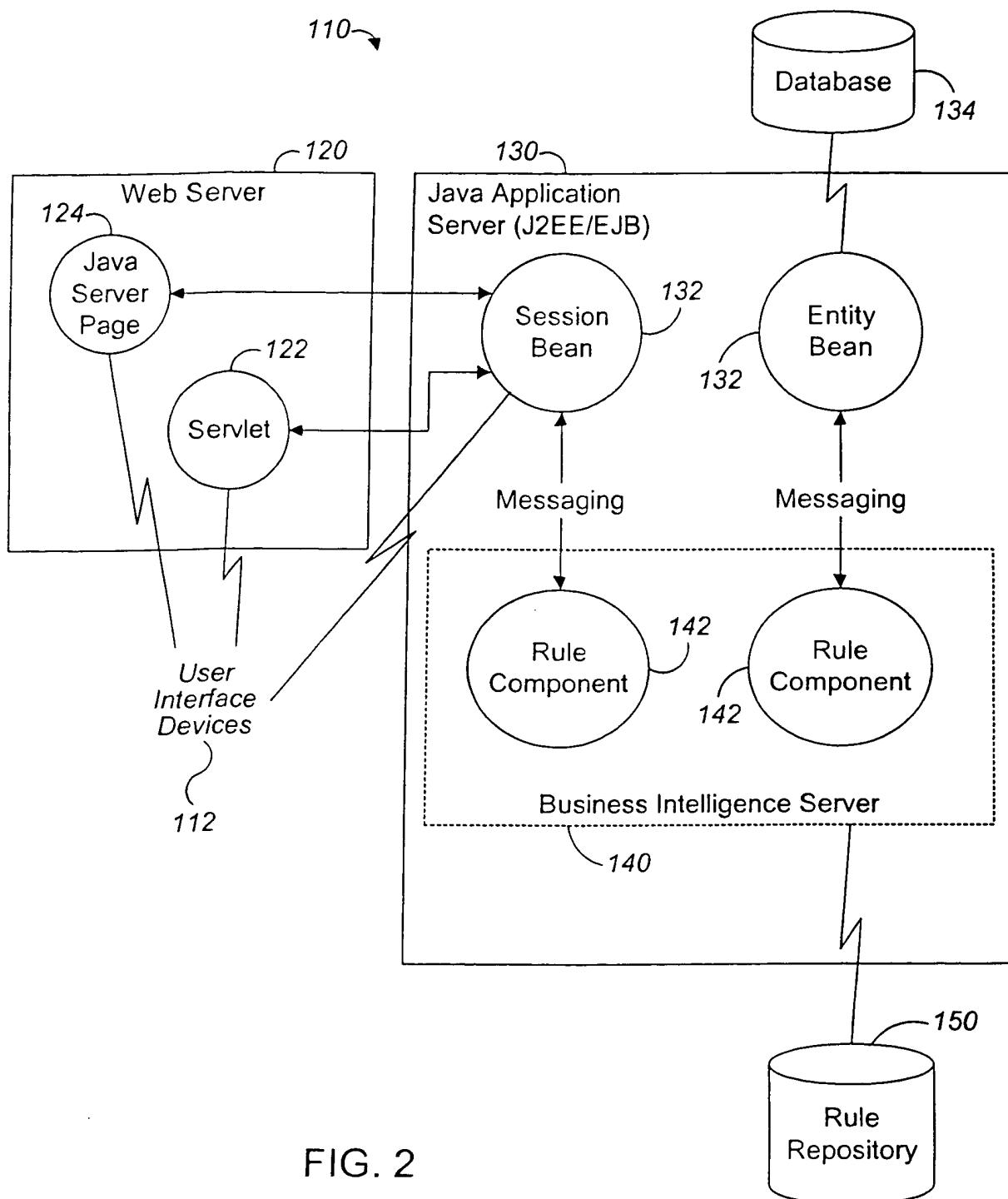
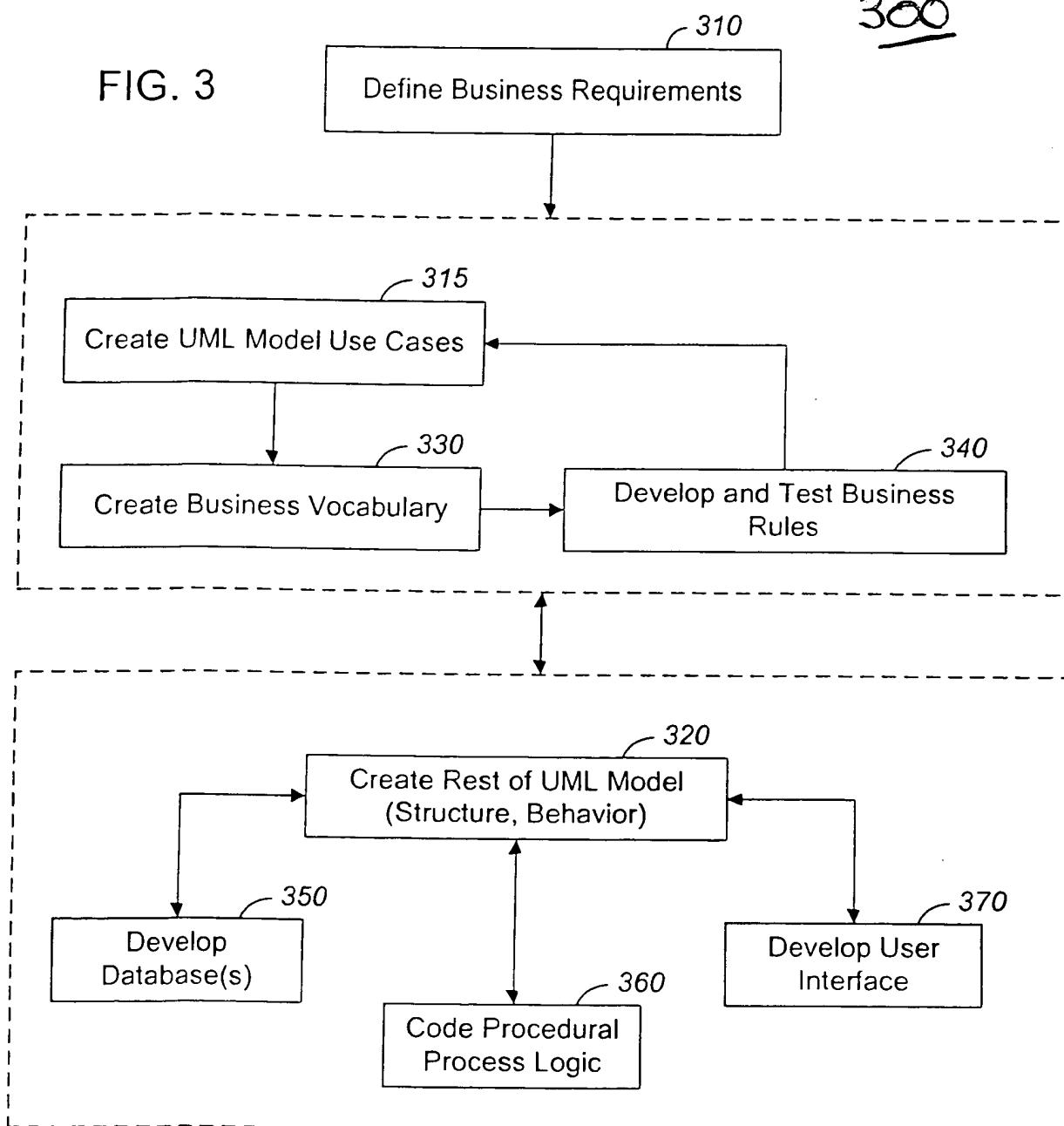


FIG. 2

FIG. 3



ProfilePerson				
Rules				
Conditions	Values	1	2	3
1 Person.smoker	{ T , F }	T	-	
2 Person.age	{ < 30 , >= 30 }	-	< 30	
3 Person.sex	{ 'M' , 'F' }	-	'F'	
4 Person.married	{ T , F }	-	T	
Actions		Values		
1 Person.risk	{ 'High' , 'Low' }	'High'	'Low'	
Overrides				
Rule Statements				
ID	Text			
1	A smoker is High risk.			
2	A married woman youner than 30 is Low risk.			

FIG. 4

420

Reject				
Rules				
	Conditions	Values	1	2
1	Person.risk	{ 'High' , 'Low' }	'High'	
2				
3				
	Actions	Values	Overrides	
1	Person.reject	{ T , F }	T	
2				
Rule Statements				
ID	Text			
1	A high risk person is rejected.			

422

FIG. 5

FIG. 6

430

ProfilePerson	
Rules	
Conditions	Values
1 Person.smoker	{T, F}
2 Person.age	{< 30, >= 30}
3 Person.sex	{'M', 'F'}
4 Person.married	{T, F}
Actions	Values
1 Person.risk	{'High', 'Low'}
Overrides	
Rule Statements	
ID	Text
1	A smoker is High risk.
2	A married woman youner than 30 is Low risk.

432

T	T
< 30	< 30
'F'	'F'
T	T
'High'	'Low'

440

ProfilePerson						
Rules						
	Conditions	Values	1	2	3	
1	Person.smoker	{ T, F }	T	-		
2	Person.age	{ < 30, >= 30 }	-	< 30		
3	Person.sex	{ 'M', 'F' }	-	'F'		
4	Person.married	{ T, F }	-	T		
	Actions	Values				
1	Person.risk	{ 'High', 'Low' }	'High'	'Low'		
Overrides						
2						
Rule Statements						
ID	Text					
1	A smoker is High risk.					
2	A married woman youner than 30 is Low risk.					

FIG. 7

Person(Mary, 20, Female, Married, Smoker)

450

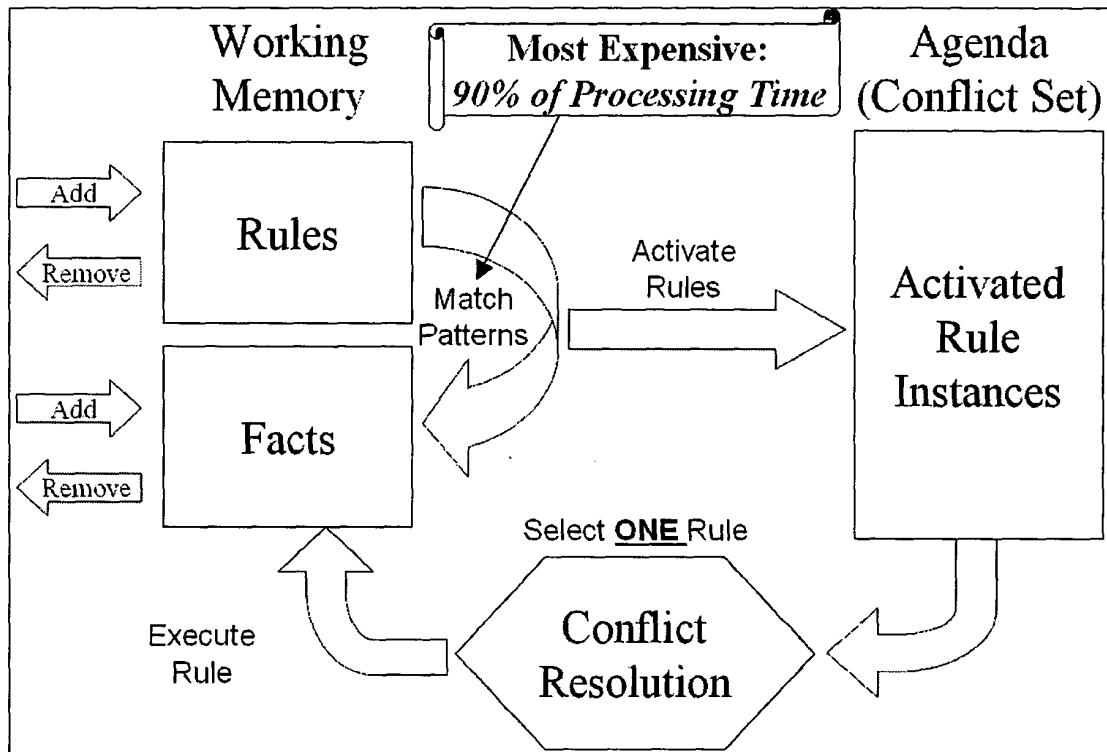
ProfilePerson						
Rules		1	2	3	4	5
1	Person.smoker	{ T, F }	T	-	F	F
2	Person.age	{ < 30, >= 30 }	-	< 30	-	>= 30
3	Person.sex	{ 'M', 'F' }	-	'F'	'M'	-
4	Person.married	{ T, F }	-	T	-	F
Actions		Values				
1	Person.risk	{ 'High', 'Low' }	'High'	'Low'		
Overrides						
Rule Statements						
ID	Text					
1	A smoker is High risk.					
2	A married woman younger than 30 is Low risk.					

FIG. 8

Person(Mary, 20, Female, Married, Smoker)

Person(Jane, 40, Female, Married, Non-smoker)

5 Other Scenarios



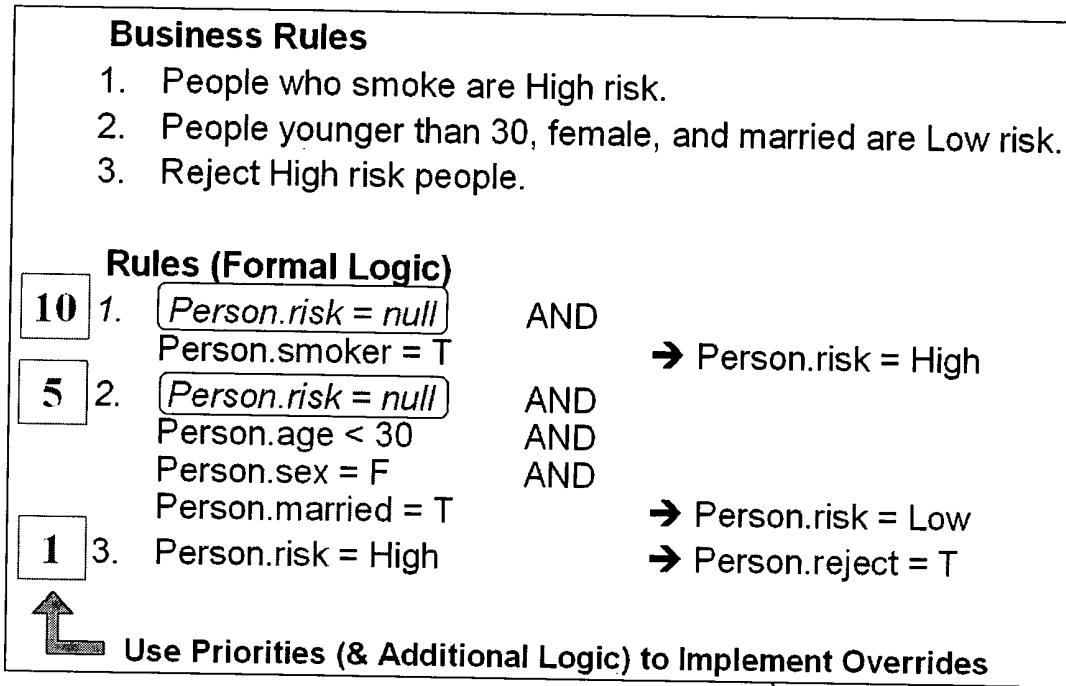


FIG. 10

470

450

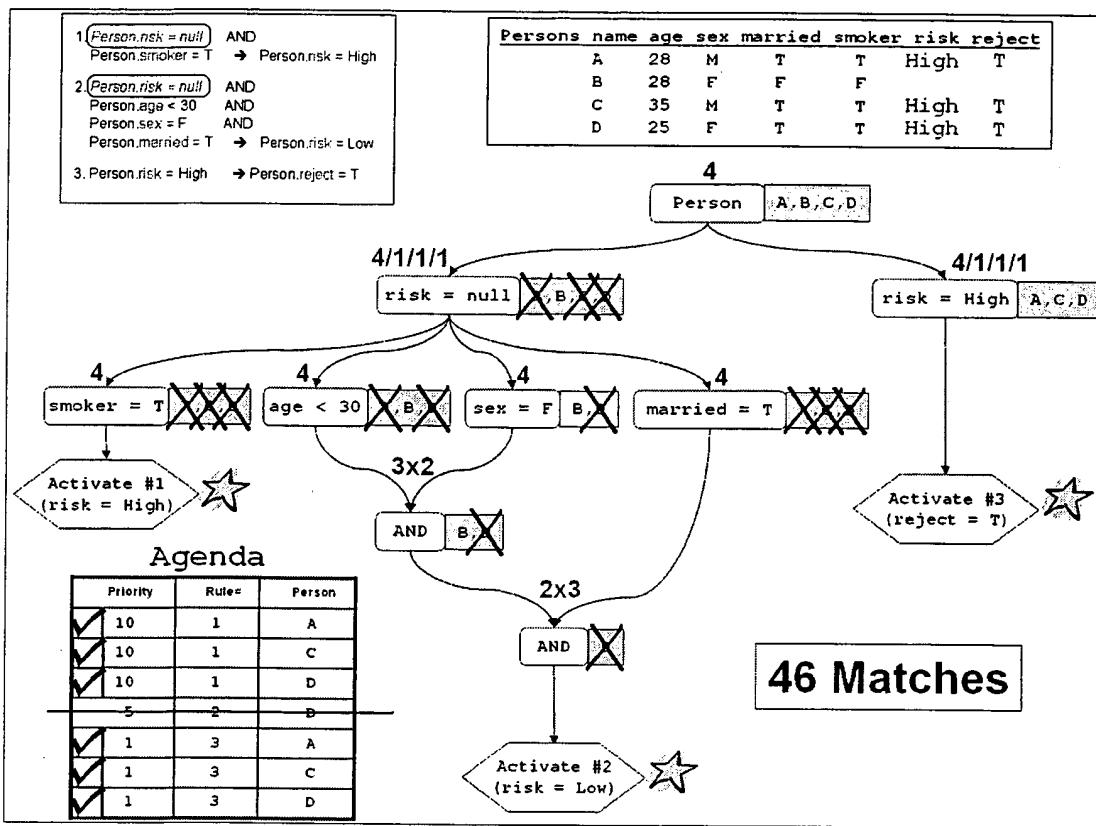


FIG. 11

480

480

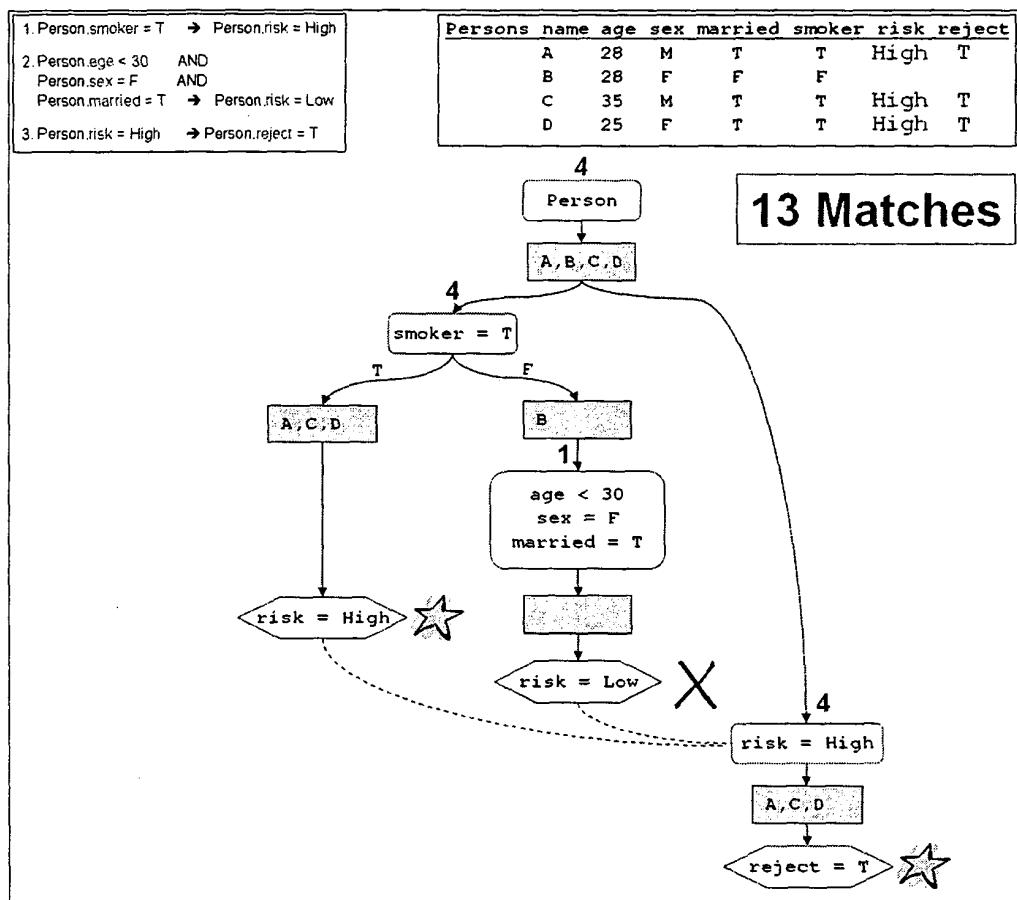


FIG. 12

490

516

Generated Business Logic

Base Library of Rule Engine

Java Virtual Machine / OS

514

JVM / OS

504

502

508

512

510

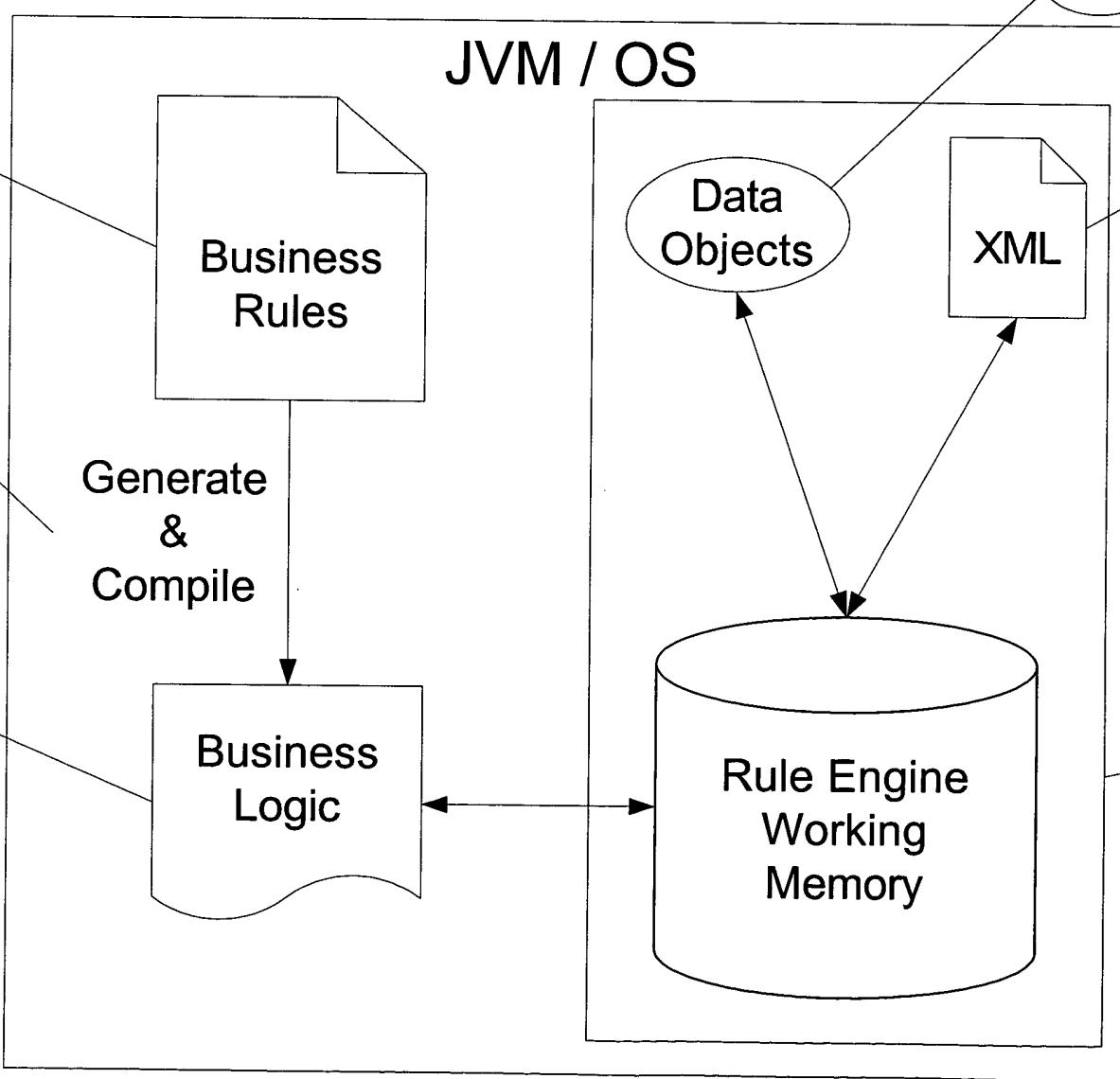


FIG. 13

```
import com.corticon.crml.*;
import com.corticon.reactor.engine.*;
import com.corticon.reactor.util.*;
import com.corticon.reactor.MaxLoopsExceededException;

class ProfilePerson implements IRuleSheetExe {
    public void execute(TupleSetManager lTSMgr)
        throws CcReactorEngineException, MaxLoopsExceededException {
        DataManager aDataMgr = lTSMgr.getDataMgr();
        boolean jumpOut;
        int runs = 0;
        int level = 1;
        boolean active = true;
        while (runs == 0 || aDataMgr.isModified()) {
            if(!active) level++;
            active=false;
            lTSMgr.genTupleSet("b", "Borrower");
            String[] lstrArr00000001 = {"b"};

            lTSMgr.joinTupleSets("ProfilePerson_nonconditional_040",lstrArr00000001);
            if (runs==0){
                lTSMgr.actOnTupleSet("ProfilePerson_nonconditional_040",
                    classProfilePerson_nonconditional_040);
                active=true;
                aDataMgr.clearWatch("ProfilePerson_nonconditional_040");
            }
            lTSMgr.genTupleSet("Loan");
            lTSMgr.extendTupleSet("Loan.borrower", "Loan", "borrower");
            String[] lstrArr00000002 = {"Loan.borrower"};

            lTSMgr.joinTupleSets("ProfilePerson_nonconditional_041",lstrArr00000002);
            if (runs==0){
                lTSMgr.actOnTupleSet("ProfilePerson_nonconditional_041",
                    classProfilePerson_nonconditional_041);
                active=true;
                aDataMgr.clearWatch("ProfilePerson_nonconditional_041");
            }
            lTSMgr.restrictTupleSet("b",
                "ProfilePerson_condition_048values_2_006",
                classProfilePerson_condition_048values_2_006);
            String[] lstrArr00000003 =
                {"ProfilePerson_condition_048values_2_006",};

            lTSMgr.unionTupleSets("ProfilePerson_rule_066ProfilePerson_condition_048",
                lstrArr00000003);
            String[] lstrArr00000004 =
                {"b", "ProfilePerson_rule_066ProfilePerson_condition_048"};
            lTSMgr.joinTupleSets("ProfilePerson_rule_066",lstrArr00000004);
            if (runs==0) {
                aDataMgr.clearWatch("ProfilePerson_rule_066");
                jumpOut = lTSMgr.actOnTupleSet("ProfilePerson_rule_066",
                    classProfilePerson_then_062);
                active=true;
            }
            if (runs==0) {
                jumpOut = lTSMgr.actOnTupleSet("ProfilePerson_rule_066",
                    classProfilePerson_then_063);
                active=true;
            }
            lTSMgr.restrictTupleSet("b",
                "ProfilePerson_condition_045values_1_004",
                classProfilePerson_condition_045values_1_004);
            String[] lstrArr00000005 =
                {"ProfilePerson_condition_045values_1_004",};

            lTSMgr.unionTupleSets("ProfilePerson_rule_061ProfilePerson_condition_045",
                lstrArr00000005);
            String[] lstrArr00000006 =
                {"b", "ProfilePerson_rule_061ProfilePerson_condition_045"};
            lTSMgr.joinTupleSets("ProfilePerson_rule_061",lstrArr00000006);
            lTSMgr.subtractTupleSet("ProfilePerson_rule_061",
                "ProfilePerson_rule_066");
            if (runs==0) {
                aDataMgr.clearWatch("ProfilePerson_rule_061");
                jumpOut = lTSMgr.actOnTupleSet("ProfilePerson_rule_061",
                    classProfilePerson_then_057);
                active=true;
            }
        }
    }
}
```

518

FIG. 14

```
if (runs==0) {
    jumpOut = lTSMgr.actOnTupleSet("ProfilePerson_rule_061",
        classProfilePerson_then_058);
    active=true;
}

lTSMgr.restrictTupleSet("b","ProfilePerson_condition_042values_0_001",
    classProfilePerson_condition_042values_0_001);
String[] lstrArr00000007 =
{"ProfilePerson_condition_042values_0_001",};

lTSMgr.unionTupleSets("ProfilePerson_rule_056ProfilePerson_condition_042",
    lstrArr00000007);
String[] lstrArr00000008 =
{"b","ProfilePerson_rule_056ProfilePerson_condition_042"};
lTSMgr.joinTupleSets("ProfilePerson_rule_056",lstrArr00000008);
lTSMgr.subtractTupleSet("ProfilePerson_rule_056",
    "ProfilePerson_rule_061");
lTSMgr.subtractTupleSet("ProfilePerson_rule_056",
    "ProfilePerson_rule_066");
if (runs==0) {
    aDataMgr.clearWatch("ProfilePerson_rule_056");
    jumpOut = lTSMgr.actOnTupleSet("ProfilePerson_rule_056",
        classProfilePerson_then_052);
    active=true;
}
if (runs==0) {
    jumpOut = lTSMgr.actOnTupleSet("ProfilePerson_rule_056",
        classProfilePerson_then_053);
    active=true;
}
runs++;
if (runs ==100)
    throw new MaxLoopsExceededException("Max Loops Exceeded");
}

class ProfilePerson_nonconditional_040 implements IAction {
    public void fire(Tuple lTuple,DataManager aDataMgr) {
        GenericEntity new_Loan =
aDataMgr.addNewEntity(lTuple,"Loan");
        GenericEntity Loan = lTuple.getEntity("Loan");
        if (Loan == null) return ;
        GenericEntity b = lTuple.getEntity("b");
        if (b == null) return ;
        b.setAssociation("loan",new_Loan);
    }
}

class ProfilePerson_nonconditional_041 implements IAction {
    public void fire(Tuple lTuple,DataManager aDataMgr) {
        GenericEntity Loan = lTuple.getEntity("Loan");
        if (Loan == null) return ;
        GenericEntity Loan_borrower =
lTuple.getEntity("Loan.borrower");
        if (Loan_borrower == null) return ;
        Loan_borrower.setAttribute("name","Eric");
    }
}

public class ProfilePerson_condition_048values_2__006 implements
ICCondition {
    public boolean test(Tuple lTuple,DataManager aDataMgr) {
        GenericEntity b = lTuple.getEntity("b");
        if (b == null) return false;
        BigDecimal b_monthlyDebt =
            (BigDecimal)b.getAttribute("monthlyDebt");
        if (b_monthlyDebt == null) return false;
        return b_monthlyDebt.compareTo(
            (Object)new BigInteger("3000"))> 0 ;
    }
}
```

FIG. 14 (cont.)

```
class ProfilePerson_then_062 implements IAction {
    public void fire(Tuple lTuple, DataManager aDataMgr) {
        GenericEntity b = lTuple.getEntity("b");
        if (b == null) return ;  

        b.setAttribute("dProfile", "noway");
    }
}  
  
class ProfilePerson_then_063 implements IAction {
    public void fire(Tuple lTuple, DataManager aDataMgr) {
        GenericEntity b = lTuple.getEntity("b");
        if (b == null) return ;  

        aDataMgr.addNewMessage("Info", "High Debtors have no
chance", b);
    }
}  
public class ProfilePerson_condition_045values_1__004 implements
ICondition {
    public boolean test(Tuple lTuple, DataManager aDataMgr) {
        GenericEntity b = lTuple.getEntity("b");
        if (b == null) return false;
        BigDecimal b_monthlyIncome =
            (BigDecimal)b.getAttribute("monthlyIncome");
        if (b_monthlyIncome == null) return false;
        return b_monthlyIncome.compareTo(
            (Object)new BigInteger("3000")) > 0 ;
    }
}  
  
class ProfilePerson_then_057 implements IAction {
    public void fire(Tuple lTuple, DataManager aDataMgr) {
        GenericEntity b = lTuple.getEntity("b");
        if (b == null) return ;
        b.setAttribute("dProfile", "low");
    }
}  
  
class ProfilePerson_then_058 implements IAction {
    public void fire(Tuple lTuple, DataManager aDataMgr) {
        GenericEntity b = lTuple.getEntity("b");
        if (b == null) return ;
        aDataMgr.addNewMessage("Info",
            "High income people are lower risk.", b);
    }
}  
  
public class ProfilePerson_condition_042values_0__001 implements
ICondition {
    public boolean test(Tuple lTuple, DataManager aDataMgr) {
        GenericEntity b = lTuple.getEntity("b");
        if (b == null) return false;
        BigInteger b_age = (BigInteger)b.getAttribute("age");
        if (b_age == null) return false;
        return b_age.compareTo((Object)new BigInteger("25")) < 0 ;
    }
}  
  
class ProfilePerson_then_052 implements IAction {
    public void fire(Tuple lTuple, DataManager aDataMgr) {
        GenericEntity b = lTuple.getEntity("b");
        if (b == null) return ;
        b.setAttribute("dProfile", "high");
    }
}  
  
class ProfilePerson_then_053 implements IAction {
    public void fire(Tuple lTuple, DataManager aDataMgr) {
        GenericEntity b = lTuple.getEntity("b");
        if (b == null) return ;
        aDataMgr.addNewMessage("Info", "Younger people are higher
risk.", b);
    }
}
```

FIG. 14 (cont.)

```
IAction classProfilePerson_nonconditional_040 = new ProfilePerson_nonconditional_040();  
  
IAction classProfilePerson_nonconditional_041 = new  
ProfilePerson_nonconditional_041();  
  
ICondition classProfilePerson_condition_048values_2_006 =  
new ProfilePerson_condition_048values_2_006();  
  
IAction classProfilePerson_then_062 = new ProfilePerson_then_062();  
  
IAction classProfilePerson_then_063 = new ProfilePerson_then_063();  
ICondition classProfilePerson_condition_045values_1_004 =  
new ProfilePerson_condition_045values_1_004();  
  
IAction classProfilePerson_then_057 = new ProfilePerson_then_057();  
  
IAction classProfilePerson_then_058 = new ProfilePerson_then_058();  
  
ICondition classProfilePerson_condition_042values_0_001 =  
new ProfilePerson_condition_042values_0_001();  
  
IAction classProfilePerson_then_052 = new ProfilePerson_then_052();  
  
IAction classProfilePerson_then_053 = new ProfilePerson_then_053();  
}
```

FIG. 14 (cont.)

```
from com.corticon.crml import OclString
from com.corticon.crml import OclDate
from com.corticon.crml import OclLiteral
from com.corticon.crml import BigInteger
from com.corticon.crml import BigDecimal
from com.corticon.reactor.engine import ICondition
from com.corticon.reactor.engine import IAction
from com.corticon.reactor.engine import IQualifier
from com.corticon.reactor.util import IRuleSheetExe
from com.corticon.reactor import MaxLoopsExceededException
from java.lang import Boolean
from java.lang import String

class ProfilePerson(IRuleSheetExe):
    def execute(self, lTSMgr):
        "@sig public void execute(com.corticon.reactor.engine.TupleSetManager lTSMgr)"
        aDataMgr = lTSMgr.getDataMgr()
        runs = 0
        level = 1
        active = 1
        while (runs == 0 or aDataMgr.isModified()):
            if(not active):
                level+=1
            active=0
            lTSMgr.genTupleSet('Person')
            lTSMgr.restrictTupleSet('Person',
                'ProfilePerson_condition_021values_0_017',
                self.classProfilePerson_condition_021values_0_017)
            lTSMgr.unionTupleSets('ProfilePerson_rule_033ProfilePerson_condition_021',
                ['ProfilePerson_condition_021values_0_017'])
            lTSMgr.joinTupleSets('ProfilePerson_rule_033',
                ['Person','ProfilePerson_rule_033ProfilePerson_condition_021'])
            if (runs==0):
                aDataMgr.clearWatch('ProfilePerson_rule_033')
                jumpOut =
            lTSMgr.actOnTupleSet('ProfilePerson_rule_033',
                self.classProfilePerson_then_030)
            active=1
            lTSMgr.restrictTupleSet('Person',
                'ProfilePerson_condition_024values_1_019',
                self.classProfilePerson_condition_024values_1_019)
            lTSMgr.restrictTupleSet('Person',
                'ProfilePerson_condition_026values_2_022',
                self.classProfilePerson_condition_026values_2_022)
            lTSMgr.restrictTupleSet('Person',
                'ProfilePerson_condition_028values_3_023',
                self.classProfilePerson_condition_028values_3_023)
            lTSMgr.unionTupleSets('ProfilePerson_rule_041ProfilePerson_condition_024',
                ['ProfilePerson_condition_024values_1_019'])
            lTSMgr.unionTupleSets('ProfilePerson_rule_041ProfilePerson_condition_026',
                ['ProfilePerson_condition_026values_2_022'])
            lTSMgr.unionTupleSets('ProfilePerson_rule_041ProfilePerson_condition_028',
                ['ProfilePerson_condition_028values_3_023'])
            lTSMgr.joinTupleSets('ProfilePerson_rule_041',
                ['Person','ProfilePerson_rule_041ProfilePerson_condition_024',
                'ProfilePerson_rule_041ProfilePerson_condition_026',
                'ProfilePerson_rule_041ProfilePerson_condition_028'])
            if (runs==0):
                aDataMgr.clearWatch('ProfilePerson_rule_041')
                jumpOut =
            lTSMgr.actOnTupleSet('ProfilePerson_rule_041',
                self.classProfilePerson_then_034)
            active=1
            runs+=1
            if(runs == 0):
                lTSMgr.clear()
            if (runs ==100):
                raise MaxLoopsExceededException('Max Loops Exceeded')
```

518

FIG. 15

```
def __init__(self):
    class ProfilePerson_condition_021values_0_017(ICondition):
        def test(this,lTuple,aDataMgr):
            "@sig public boolean test(
                com.corticon.reactor.engine.Tuple lTuple,
                com.corticon.reactor.engine.DataManager aDataMgr)"
            Person = lTuple.getEntity('Person')
            if Person == None : return 0
            Person_smoker = Person.getAttribute('smoker')
            if Person_smoker == None : return 0
            return Person_smoker
    self.classProfilePerson_condition_021values_0_017=
        ProfilePerson_condition_021values_0_017()

    class ProfilePerson_then_030(IAction):
        def fire(this,lTuple,aDataMgr):
            "@sig public void fire(
                com.corticon.reactor.engine.Tuple
                lTuple,com.corticon.reactor.engine.DataManager
                aDataMgr)"
            Person = lTuple.getEntity('Person')
            if Person == None : return None

            Person_risk = Person.setAttribute('risk','High')

    self.classProfilePerson_then_030=ProfilePerson_then_030()
    class ProfilePerson_condition_024values_1_019(ICondition):
        def test(this,lTuple,aDataMgr):
            "@sig public boolean test(
                com.corticon.reactor.engine.Tuple lTuple,
                com.corticon.reactor.engine.DataManager aDataMgr)"
            Person = lTuple.getEntity('Person')
            if Person == None : return 0
            Person_age = Person.getAttribute('age')
            if Person_age == None : return 0
            return Person_age.compareTo(BigInteger('30'))< 0
    self.classProfilePerson_condition_024values_1_019=
        ProfilePerson_condition_024values_1_019()

    class ProfilePerson_condition_026values_2_022(ICondition):
        def test(this,lTuple,aDataMgr):
            "@sig public boolean test(
                com.corticon.reactor.engine.Tuple lTuple,
                com.corticon.reactor.engine.DataManager aDataMgr)"
            Person = lTuple.getEntity('Person')
            if Person == None : return 0
            Person_sex = Person.getAttribute('sex')
            if Person_sex == None : return 0
            return Person_sex=='F'
    self.classProfilePerson_condition_026values_2_022=
        ProfilePerson_condition_026values_2_022()

    class ProfilePerson_condition_028values_3_023(ICondition):
        def test(this,lTuple,aDataMgr):
            "@sig public boolean test(
                com.corticon.reactor.engine.Tuple lTuple,
                com.corticon.reactor.engine.DataManager aDataMgr)"
            Person = lTuple.getEntity('Person')
            if Person == None : return 0
            Person_married = Person.getAttribute('married')
            if Person_married == None : return 0
            return Person_married
    self.classProfilePerson_condition_028values_3_023=
        ProfilePerson_condition_028values_3_023()
```

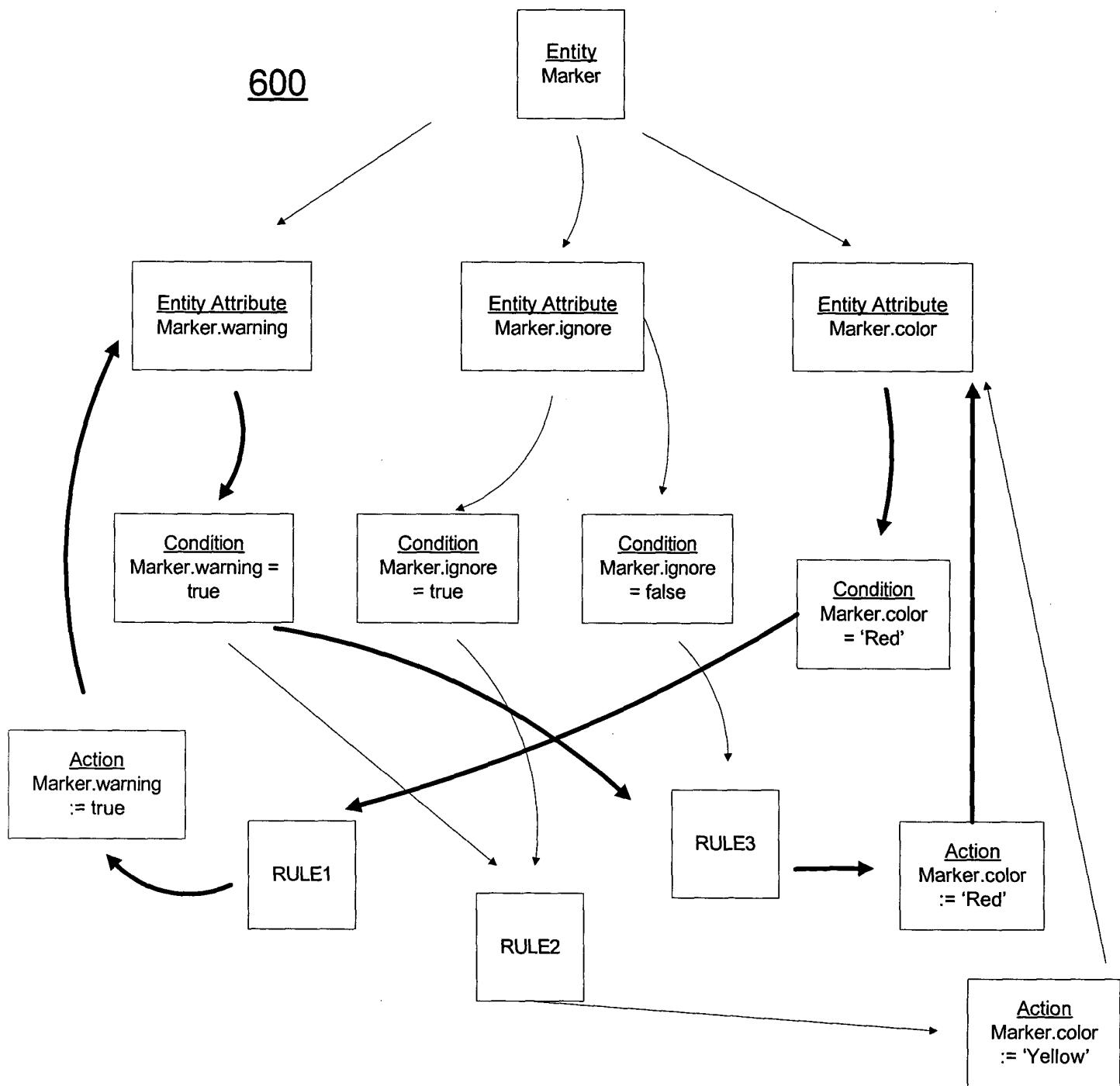
FIG. 15 (cont.)

```
class ProfilePerson_then_034(IAction):
    def fire(this,lTuple,aDataMgr):
        "@sig public void fire(
            com.corticon.reactor.engine.Tuple lTuple,
            com.corticon.reactor.engine.DataManager aDataMgr)"
        Person = lTuple.getEntity('Person')
        if Person == None : return None
        Person._risk = Person.setAttribute('risk','Low')

self.classProfilePerson_then_034=ProfilePerson_then_034()
self.ok = 1
```

FIG. 15 (cont.)

- [RULE1] IF Marker.color = 'Red' THEN Marker.warning := true
- [RULE2] IF Marker.warning = true AND Marker.ignore = true THEN Marker.color := 'Yellow'
- [RULE3] IF Marker.warning = true AND Marker.ignore = false THEN Marker.color := 'Red'



CONVEX SUBSET:

LOGICAL LOOP 1 (edges in BOLD)= color : RULE1 : warning : RULE2 : color
 LOGICAL LOOP 2 = color : RULE1 : warning : RULE3 : color

FIG. 16

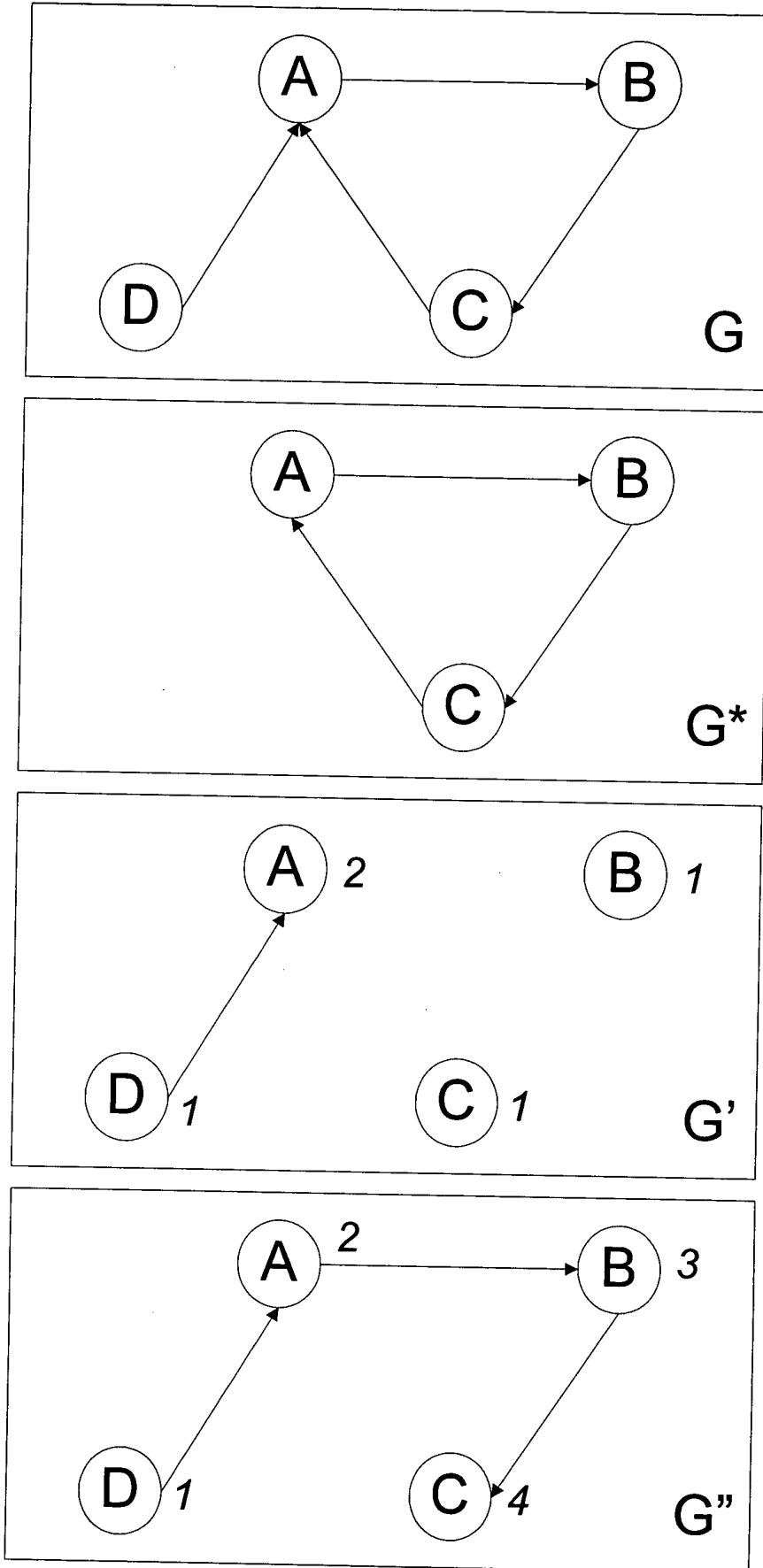


FIG. 17